What is claimed is:

1. A computer-implemented method of debugging an object-oriented computer program, the method comprising:

(a) in response to user input, setting an inheritance breakpoint that is associated with a first program entity in the object-oriented computer program in which is identified a method; and

(b) halting execution of the object-oriented computer program during debugging in response to reaching an implementation of the method defined in a second program entity in the object-oriented computer program that is different from the first program entity.

2. The computer-implemented method of claim 1, wherein the first program entity is an interface that identifies the method, and wherein the second program entity is a class that implements the method.

3. The computer-implemented method of claim 1, wherein the first program entity is a first class that includes a second implementation of the method, wherein the second program entity is a second class that inherits from the first class, and wherein the first implementation of the method in the second class overrides the second implementation of the method in the first class.

4. The computer-implemented method of claim 3, wherein the second class is a subclass of the first class.

5. The computer-implemented method of claim 1, wherein the first program entity is an abstract class that identifies the method, and wherein the second program entity is a non-abstract class that implements the method.

6. The computer-implemented method of claim 1, wherein the inheritance breakpoint is additionally associated with the method.

7. The computer-implemented method of claim 6, wherein setting the inheritance breakpoint includes storing in a breakpoint data structure an entry that identifies the first program entity and the method.

8. The computer-implemented method of claim 1, further comprising, during loading of a class in the object-oriented computer program, identifying each implementation of the method in the class and setting a breakpoint on such implementation, wherein halting execution of the object-oriented computer program during debugging in response to reaching the implementation of the method includes reaching a breakpoint set on such implementation.

9. The computer-implemented method of claim 1, further comprising setting a breakpoint on each implementation of the method, wherein halting execution of the object-oriented computer program during debugging in response to reaching the implementation of the method includes reaching a breakpoint set on such implementation.

10. The computer-implemented method of claim 9, wherein setting a breakpoint on each implementation of the method includes setting a breakpoint on a first statement in an implementation of the method.

11. The computer-implemented method of claim 9, wherein setting a breakpoint on each implementation of the method includes setting a breakpoint on a method call to an implementation of the method.

12. The computer-implemented method of claim 1, wherein setting the inheritance breakpoint includes associating a user-specified condition with the inheritance breakpoint, and wherein halting execution of the object-oriented computer program during debugging in response to reaching the implementation of the method is performed only if the user-specified condition has been met.

13. A computer-implemented method of debugging an object-oriented computer program, the method comprising:

(a) in response to user input, setting an inheritance breakpoint that is associated with a first class in the object-oriented computer program in which is identified a method; and

(b) halting execution of the object-oriented computer program during debugging in response to reaching an implementation of the method defined in a second class in the object-oriented computer program that inherits from the first class.

14. A computer-implemented method of debugging an object-oriented computer program, the method comprising:

(a) in response to user input, setting an inheritance breakpoint that is associated with an interface in the object-oriented computer program in which is identified a method; and

(b) halting execution of the object-oriented computer program during debugging in response to reaching an implementation of the method defined in a class in the object-oriented computer program that implements the interface.

15. A computer-implemented method of debugging an object-oriented computer program, the method comprising:

(a) receiving user input to halt program execution during debugging in response to reaching any of a plurality of implementations of a method in an object-oriented computer program; and

(b) thereafter setting a breakpoint for at least a subset of the plurality of implementations such that execution of the object-oriented computer program will be halted in response to reaching any of the implementations on which a breakpoint has been set.

16. The computer-implemented method of claim 15, wherein the user input to halt program execution includes user input to set an inheritance breakpoint on the method.

17. The computer-implemented method of claim 15, wherein setting the breakpoint includes, during loading of a class in the object-oriented computer program, identifying each implementation of the method in the class and setting a breakpoint on such implementation.

18. An apparatus, comprising:

(a) a memory within which is resident at least a portion of an object-oriented computer program under debug, the object-oriented computer program including a first program entity in which is identified a method, and a second program entity that is different from the first program entity, and that includes an implementation of the method; and

(b) program code configured to set an inheritance breakpoint that is associated with the first program entity in response to user input, and to halt execution of the object-oriented computer program during debugging in response to reaching the implementation of the method defined in the second program entity.

19. The apparatus of claim 18, wherein the first program entity is an interface that identifies the method, and wherein the second program entity is a class that implements the method.

20. The apparatus of claim 18, wherein the first program entity is a first class that includes a second implementation of the method, wherein the second program entity is a second class that inherits from the first class, and wherein the first implementation of the method in the second class overrides the second implementation of the method in the first class.

21. The apparatus of claim 18, wherein the first program entity is an abstract class that identifies the method, and wherein the second program entity is a non-abstract class that implements the method.

22. The apparatus of claim 18, wherein the inheritance breakpoint is additionally associated with the method, and wherein the program code is configured to store in a breakpoint data structure an entry that identifies the first program entity and the method.

23. The apparatus of claim 18, wherein the program code is further configured to set a breakpoint on each implementation of the method, and wherein the program code is configured to halt execution of the object-oriented computer program during debugging in response to reaching a breakpoint set on such implementation.

24. The apparatus of claim 23, wherein the program code is configured to set the breakpoint on each implementation of the method by dynamically setting a breakpoint on each implementation of the method in a class in the object-oriented computer program during loading of the class.

25. The apparatus of claim 18, wherein the program code is configured to associate a user-specified condition with the inheritance breakpoint, and wherein the program code is configured to halt execution of the object-oriented computer program during debugging in response to reaching the implementation of the method only if the user-specified condition has been met.

26. An apparatus, comprising:

(a) a memory within which is resident at least a portion of an object-oriented computer program under debug, the object-oriented computer program including a method and a plurality of implementations of the method; and

(b) program code configured to receive user input to halt program execution during debugging in response to reaching any of the plurality of implementations of the method, and to thereafter set a breakpoint for at least a subset of the plurality of implementations such that execution of the object-oriented computer program will be halted in response to reaching any of the implementations on which a breakpoint has been set.

27. The apparatus of claim 26, wherein the user input to halt program execution includes user input to set an inheritance breakpoint on the method, and wherein the program code is configured to set a breakpoint by, during loading of a class in the object-oriented computer program, identifying each implementation of the method in the class and setting a breakpoint on such implementation.

28. A program product, comprising:

(a) program code configured to set an inheritance breakpoint in response to user input, wherein the inheritance breakpoint is associated with a first program entity in an object-oriented computer program in which is identified a method, and to halt execution of the object-oriented computer program during debugging in response to reaching an implementation of the method defined in a second program entity in the object-oriented computer program that is different from the first program entity; and

(b) a signal bearing medium bearing the program code.

29. The program product of claim 28, wherein the signal bearing medium includes at least one of a transmission medium and a recordable medium.

30. A program product, comprising:

(a) program code configured to receive user input to halt program execution of an object-oriented computer program during debugging in response to reaching any of a plurality of implementations of a method in the object-oriented computer program, and to thereafter set a breakpoint for at least a subset of the plurality of implementations such that execution of the object-oriented computer program will be halted in response to reaching any of the implementations on which a breakpoint has been set; and

(b) a signal bearing medium bearing the program code.